H3C OAP Module

Developing Manual

Hangzhou H3C Technologies Co., Ltd.

http://www.h3c.com

Manual Version: 20070723-C-1.01

Copyright © 2006-2007, Hangzhou H3C Technologies Co., Ltd. and its licensors

All Rights Reserved

No part of this manual may be reproduced or transmitted in any form or by any means without prior written consent of Hangzhou H3C Technologies Co., Ltd.

Trademarks

H3C, H3C, Aolynk, Aolynk, H^{3}Care, H3care, TOP G, TOPG, IRF, NetPilot, Neocean, NeoVTL, SecPro, SecPoint, SecEngine, SecPath, Comware, Secware, Storware, NQA, VVG, V^{2}G, V^{n}G, PSPT, XGbus, N-Bus, TiGem, InnoVision and HUASAN are trademarks of Hangzhou H3C Technologies Co., Ltd.

All other trademarks that may be mentioned in this manual are the property of their respective owners.

Notice

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied.

To obtain the latest information, please access:

http://www. h3c.com

Technical Support

customer_service@h3c.com

http://www. h3c.com

About This Manual

Related Documentation

In addition to this manual, each H3C OAP Series Module documentation set includes the following:

Manual	Description
H3C OAP Module User Manual	It introduces the OAP/OAPS module's application.

Organization

H3C OAP Module Developing Manual is organized as follows:

Chapter	Contents
1 Operation Guide for OAP Module Customization	It introduces the additional developing issues for OAP module's application.
2 Operation Guide for OAPS Module Customization	It introduces the additional developing issues for OAPS module's application.

Conventions

The manual uses the following conventions:

I. Command conventions

Convention	Description
Boldface	The keywords of a command line are in **Boldface**.
italic	Command arguments are in *italic*.
[ ]	Items (keywords or arguments) in square brackets [ ] are optional.
{ x | y | ... }	Alternative items are grouped in braces and separated by vertical bars. One is selected.
[x | y | ...]	Optional alternative items are grouped in square brackets and separated by vertical bars. One or none is selected.

Convention	Description
{ x | y | ... } *	Alternative items are grouped in braces and separated by vertical bars. A minimum of one or a maximum of all can be selected.
[x | y | ...] *	Optional alternative items are grouped in square brackets and separated by vertical bars. Many or none can be selected.
&<1-n>	The argument(s) before the ampersand (&) sign can be entered 1 to n times.
#	A line starting with the # sign is comments.

II. Symbols

Convention	Description
Warning	Means reader be extremely careful. Improper operation may cause bodily injury.
Caution	Means reader be careful. Improper operation may cause data loss or damage to equipment.
🕮 Note	Means a complementary description.

Environmental Protection

This product has been designed to comply with the requirements on environmental protection. For the proper storage, use and disposal of this product, national laws and regulations must be observed.

Table of Contents

Chapter 1 Operation Guide for OAP Module Customization 1-1
1.1 Booting the Linux OS on an OAP Module 1-1
1.2 Compiling the Linux Kernel and the Module 1-2
Chapter 2 Operation Guide for OAPS Module Customization 2-1
2.1 Booting the Linux OS on an OAPS Module 2-1
2.2 Compiling the Linux Kernel and the Module 2-2

Chapter 1 Operation Guide for OAP Module Customization

1.1 Booting the Linux OS on an OAP Module

OAP modules support two storage media: CF card (required) and hard disk (optional). On the CF card stored the master boot recorder (MBR) installed with LILO and in addition the kernel and initrd files needed for booting the Linux OS.

Note that the OAP module uses the Linux Loader (LILO) rather than the grand unified bootloader (GRUB) as the bootloader.

When LILO starts, you can see the following:

```
============<BIOS-BASIC MENU(Ver 1.13)>================
| <1> Boot Linux From CF  Card                         |
| <2> Install Linux OS  With Ethernet                  |
| <3> Backup BIOS to the backup portion                |
| <4> Recover BIOS from the backup portion             |
| <5> Reboot                                           |
=======================================================
Enter your choice(1-5):1Reading boot sector...done.
Switching to boot loader.


LILO boot:
```

At the prompt, press **Tab**. You can see the following two boot options:

```
linux-hdc       linux-ramdisk
boot:
```

Enter either option to boot the corresponding system.

The following describes the two options in detail.

I. Linux-ramdisk

If the linux-ramdisk option is selected, ramdisk will be used as the root file system and the first partition of the CF card will be mounted as the /boot partition. The system provides such functions as private protocol module, dongle and CPLD driver.

II. Linux-hdc

If the linux-hdc option is selected, LILO uses the hard disk as the root file system to boot the Linux OS. A complete root file system is installed on the hard disk, providing tools such as GCC, RPM and VI Editor and some network services. You can configure the

services on the Linux OS just as you do on a PC. By default, the linux-hdc option is selected.

By default, the hard disk and CF card are partitioned as follows:

Device name	Partition name	Mount point	File system format	Function
CF card	/dev/hda1	/boot	ext3	The boot partition for the Linux OS. LILO is stored in MBR. In addition, the kernel and initrd files needed for booting the Linux OS are stored on the card. The volume label of the partition is /boot.
Hard disk	/dev/hdc1	/	ext3	The root file system for the Linux OS when the linux-hdc option is selected. The volume label of the partition is /.
	/dev/hdc2	/alt	ext3	Backup partition of the root file system. Its factory setting is null.
	/dev/hdc3	/opt	ext3	The largest partition for users. No system file is stored in it.
	/dev/hdc4	swap	swap	The swap partition for the Linux OS when the linux-hdc option is selected.

You can customize partitions as needed, except that only the first partition of the CF card can be as the boot partition.

1.2 Compiling the Linux Kernel and the Module

If you select linux-hdc, the system takes /dev/hdc1 of the hardware as the root file system. You can then customize the OAP module based on the Linux OS. This is the same as you do on a PC running the Linux OS, except that the input/output mode is different.

I. Compiling the Linux kernel

Sometimes, you may need to recompile the kernel to accommodate requirements such as reconfiguring features of Linux kernel or compiling some loadable device driver.

The OAP module is not delivered with the 2.6.9-42.0.3.EL source codes. You can download the source code from the following web site:

ftp://updates.redhat.com/enterprise/4AS/en/os/SRPMS/kernel-2.6.9-42.0.3.EL.src.rpm

To use the code for the OAP module, you need to install the following patch program in addition:

```
kernel-2.6.9-42.0.3.EL-appboard.patch
```

You are recommended to compile the Linux kernel in the following procedure:

1) Download the kernel source code to the directory /opt.

```
[root@localhost root]#cd /opt
[root@localhost                                                    opt]#scp
root@192.168.0.2:/home/zxh/centos/kernel-2.6.9-42.0.3.EL.src.rpm ./
root@192.168.0.2's password:
kernel-2.6.9-42.0.3.EL.src.rpm           100%   45MB  11.3MB/s   00:04
[root@localhost opt]#
```

2) Download the patch.

```
[root@localhost                          opt]#                          scp
root@192.168.0.2:/home/zxh/centos/kernel-2.6.9-42.0.3.EL-appboard.patch ./
root@192.168.0.2's password:
kernel-2.6.9-42.0.3.EL-appboard.patch    100%   14KB  13.8KB/s   00:00
[root@localhost opt]#
```

3) Decompress the source code of the kernel.

```
[root@localhost opt]# rpm -Uvh kernel-2.6.9-42.0.3.EL.src.rpm
[root@localhost opt]# cd /usr/src/redhat/SPECS/
[root@localhost SPECS]# rpmbuild -bp --target $(uname -m) kernel-2.6.spec
```

4) Patch the initial source code, generating the kernel source code applicable to the OAP module.

```
[root@localhost SPECS]#cd ../BUILD/kernel-2.6.9/linux-2.6.9/
[root@localhost                                              linux-2.6.9]#cp
/opt/kernel-2.6.9-42.0.3.EL-appboard.patch ./
[root@localhost         linux-2.6.9]#         patch          -p1         <
kernel-2.6.9-42.0.3.EL-appboard.patch
patching file arch/i386/boot/video.S
patching file arch/i386/kernel/reboot.c
patching file drivers/net/e1000/e1000_hw.c
patching file drivers/serial/8250.c
patching file init/main.c
patching file Makefile
[root@localhost linux-2.6.9]#
```

5) Compile the kernel and driver.

```
[root@localhost linux-2.6.9]# make clean
[root@localhost linux-2.6.9]# make mrproper
[root@localhost linux-2.6.9]# cp configs/kernel-2.6.9-i686.config ./.config
[root@localhost linux-2.6.9]# make oldconfig
```

```
[root@localhost linux-2.6.9]# make menuconfig
```

Customize the features of the kernel as needed.

```
[root@localhost linux-2.6.9]# make
[root@localhost linux-2.6.9]# make modules
[root@localhost linux-2.6.9]# make modules_install
[root@localhost linux-2.6.9]# make installl
```

II. Configuring LILO

After compiling the kernel and initrd, manually update LILO so that it can use the new kernel to load the Linux OS.

You can use the VI Editor to modify the startup settings in the LILO configuration file lilo.conf, which is available under /etc. The following are the sample contents of lilo.conf:

```
prompt
timeout=50
default=linux-hdc
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
linear
image=/boot/vmlinuz-2.6.9-42.0.3.EL
         label=linux-ramdisk
         initrd=/boot/initrd-2.6.9-42.0.3.EL-appboard.img
         append="root=/dev/ram0 console=ttyS0,9600 rw"
image=/boot/vmlinuz-2.6.9-42.0.3.EL
         label=linux-hdc
         initrd=/boot/initrd-2.6.9-42.0.3.EL.img
         append="console=ttyS0,9600 root=LABEL=/ "
```

Because the startup information of the OAP module is input/output through a serial port, you need to append the following configuration through LILO:

```
append="console=ttyS0,9600"
```

Suppose you are using the HyperTerminal for configuration. On the HyperTerminal, set terminal emulation to VT100, connect to the console port on your network device, and enter the **oap connect slot x** command to connect to the serial port on the OAP module. The VI Editor can thus work normally. The following figure shows the settings that you should make on the HyperTerminal.

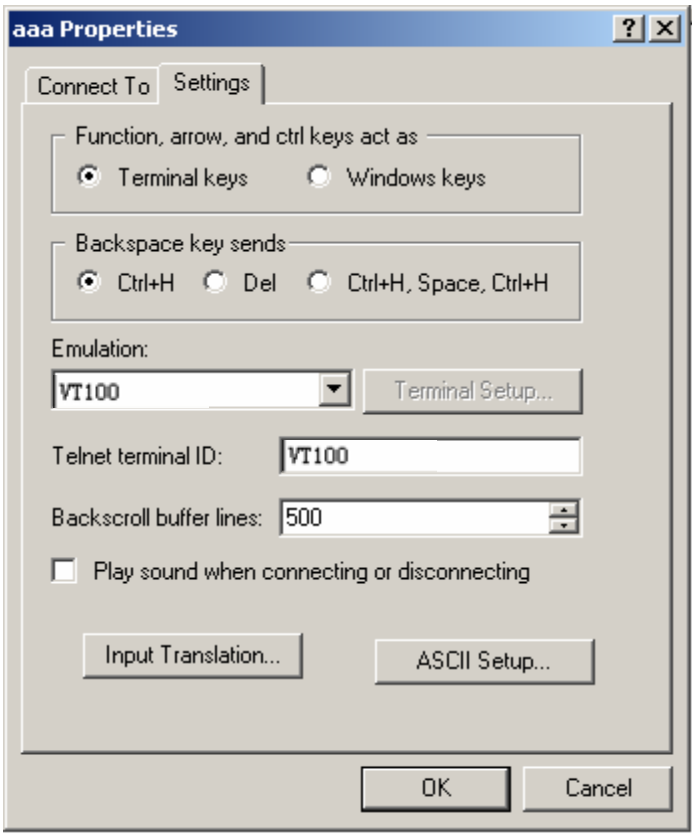

Figure 1-1 Setting terminal type

III. Compiling e1000 driver

An Intel Gigabit Ethernet chip is used on the OAP module and the default e1000 driver is 7.0.33. The kernel patching operation mentioned earlier modifies the e1000 codes as well. To update the Ethernet driver, e1000 drive 7.0.41 for example, do the following:

1) Download the source code e1000-7.0.41.tar.gz of e1000 driver 7.0.41 from the official Web of Intel and then decompress it.

```
[root@localhost tmp]# ls
e1000-7.0.41.tar.gz
[root@localhost tmp]# tar -zxf e1000-7.0.41.tar.gz -C ./
[root@localhost tmp]#
```

2) Obtain the patch file for e1000 driver and copy it to a specific directory.

```
[root@localhost tmp]# cd e1000-7.0.41
[root@localhost e1000-7.0.41]# ls
e1000.7                      e1000.spec    LICENSE      README  SUMS
e1000-7.0.41-appboard.patch  ldistrib.txt  pci.updates  src
[root@localhost e1000-7.0.41]# patch -p1 < e1000-7.0.41-appboard.patch
patching file src/e1000.h
patching file src/e1000_hw.c
```

```
patching file src/e1000_main.c
[root@localhost e1000-7.0.41]#
```

3) Compile e1000 driver.

```
[root@localhost e1000-7.0.41]#cd src
[root@localhost src]#make clean
[root@localhost src]#make
[root@localhost src]#make install
```

4) Reload the system and update e1000 to a specified version.

```
[root@localhost root]# ethtool -i eth0
driver: e1000
version: 7.0.41
firmware-version: N/A
bus-info: 01:00.0
[root@localhost root]#
```

You may compile other modules following the usual compilation procedure.

Chapter 2 Operation Guide for OAPS Module Customization

2.1 Booting the Linux OS on an OAPS Module

An OAPS module provides two storage media: CF card and hard disk. LILO is installed On the CF card stored the MBR installed with LILO and the kernel and initrd files needed for booting the Linux OS are saved on the CF card.

Note that the OAPS module supports both LILO and GRUB as the bootloader of the system.

When LILO starts, you can see the following:

```
============<BIOS-BASIC MENU(Ver1.00)>=================
| <1> Boot Linux from CF Card                          |
| <2> Boot Linux from Hard Disk                        |
| <3> Boot Linux from USB                              |
| <4> Boot Linux from Ethernet                         |
| <5> Default boot device                              |
| <6> Backup BIOS to the backup portion                |
| <7> Recover BIOS from the backup portion             |
| <8> Reboot                                           |
=======================================================
Enter your choice(1-8):1
Reading boot sector...done.
Switching to boot loader.


LILO boot:
```

At the prompt, press **Tab**. you can see the following two boot options:

```
linux-hdc       linux-ramdisk
boot:
```

Enter either option to boot the corresponding system.

The following describes the two options in detail.

I. Linux-ramdisk

If the linux-ramdisk option is selected, ramdisk will be used as the root file system and the first partition of the CF card will be mounted as the /boot partition. The system provides such functions as private protocol module, dongle and CPLD driver.

II. Linux-hdc

If the linux-hdc option is selected, LILO uses the hard disk as the root file system to boot the Linux OS. A complete root file system is installed on the hard disk, providing tools such as GCC, RPM and VI Editor and some network services. You can configure the services on the Linux OS just as you do on a PC. By default, the linux-hdc option is selected.

By default, the hard disk and CF card are partitioned as follows:

Device name	Partition name	Mount point	File system format	Function
CF card	/dev/hda1	/boot	ext3	The boot partition for the Linux OS. LILO is stored in MBR. In addition, the kernel and initrd files needed for booting the Linux OS are stored on the card. The volume label of the partition is /boot.
Hard disk	/dev/sda1	/	ext3	The root file system for the Linux OS when the linux-hdc option is selected. The volume label of the partition is /.
	/dev/sda2	/alt	ext3	Backup partition of the root file system. Its factory setting is null.
	/dev/sda3	/opt	ext3	The largest partition for users. No system file is stored in it.
	/dev/sda4	swap	swap	The swap partition for the Linux OS when the linux-hdc option is selected.

You can customize partition mode as needed.

2.2 Compiling the Linux Kernel and the Module

If you select the linux-hdc option when booting from the CF card or select to boot from the hard disk, the system takes the /dev/sda1 partition of the hardware as the root file system. You can then customize the OAPS module based on the Linux OS. This is the same as you do on a PC running the Linux OS, except that the input/output mode is different.

I. Compiling the Linux kernel

To recompile the kernel to accommodate special functions, you can download the kernel from the following web site:

ftp://updates.redhat.com/enterprise/4AS/en/os/SRPMS/kernel-2.6.9-42.0.3.EL.src.rpm

After downloading the kernel, follow these steps to decompress and compile the kernel (assuming the source codes of the kernel have been downloaded to the directory /opt):

1) Decompress the source codes of the kernel.

```
[root@localhost root]#cd /opt
[root@localhost opt]# rpm -Uvh kernel-2.6.9-42.0.3.EL.src.rpm
[root@localhost opt]# cd /usr/src/redhat/SPECS/
[root@localhost SPECS]# rpmbuild -bp --target $(uname -m) kernel-2.6.spec
```

2) Configure the kernel.

```
[root@localhost linux-2.6.9]# make clean
[root@localhost linux-2.6.9]# make mrproper
[root@localhost linux-2.6.9]# make menuconfig
```

3) Customize the features of the kernel as needed.

```
[root@localhost linux-2.6.9]# make
[root@localhost linux-2.6.9]# make modules
[root@localhost linux-2.6.9]# make modules_install
[root@localhost linux-2.6.9]# make installl
```

II. Configuring LILO

After compiling the kernel and initrd, manually update LILO so that it can use the new kernel to load the Linux OS.

You can use the VI Editor to modify the startup settings in the LILO configuration file lilo.conf, which is available under /etc. The following are the sample contents of lilo.conf:

```
prompt
timeout=50
default=linux-hdc
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
linear
image=/boot/vmlinuz-2.6.9-42.0.3.EL-oaps
         label=linux-ramdisk
         initrd=/boot/initrd-2.6.9-42.0.3.EL-appboard-oaps.img
         append="root=/dev/ram0 console=ttyS0,9600 rw"
image=/boot/vmlinuz-2.6.9-42.0.3.EL-oaps
         label=linux-hdc
         initrd=/boot/initrd-2.6.9-42.0.3.EL-oaps.img
```

```
append="console=ttyS0,9600 rw root=LABEL=/"
```

Because the startup information of the OAPS module is input/output through a serial port, you need to append the following configuration through LILO:

```
append="console=ttyS0,9600"
```

Suppose you are using the HyperTerminal for configuration. On the HyperTerminal, set terminal emulation to VT100, connect to the console port on your network device, and enter the **oap connect slot x** command to connect to the serial port on the OAPS module. The VI Editor can thus work normally. The following figure shows the settings that you should make on the HyperTerminal.

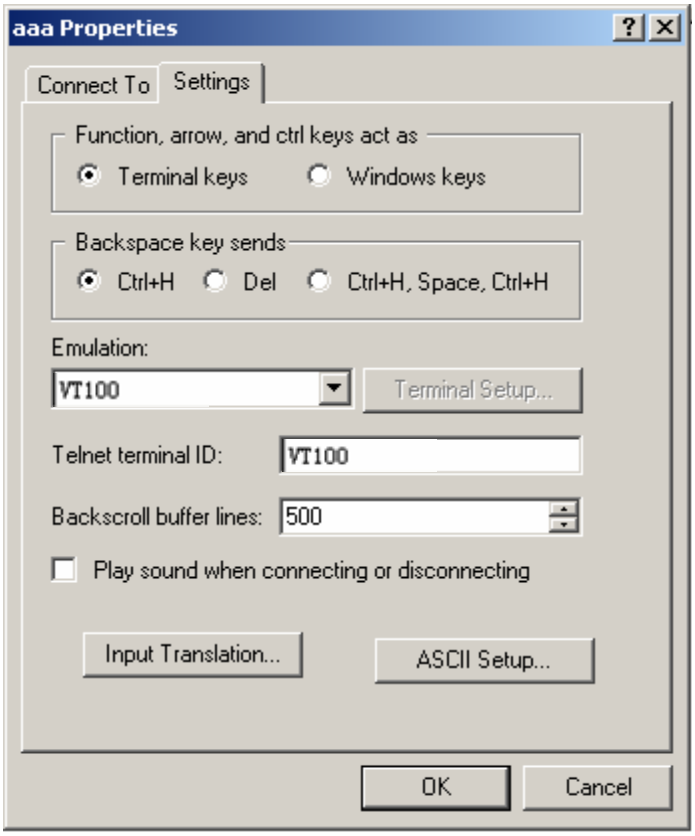

Figure 2-1 Setting terminal type

H3C OAP 模块 开发手册

H3C OAP 模块

开发手册

杭州华三通信技术有限公司

http://www.h3c.com.cn

资料版本：20070723-C-1.01

声明

Copyright ©2006-2007 杭州华三通信技术有限公司及其许可者 版权所有，保留一切权利。

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本书内容的部分或全部，并不得以任何形式传播。

H3C、H3C、Aolynk、Aolynk、H^{3}Care、H3care、TOP G、TOPG、IRF、NetPilot、Neocean、NeoVTL、SecPro、SecPoint、SecEngine、SecPath、Comware、Secware、Storware、NQA、VVG、V^{2}G、V^{n}G、PSPT、XGbus、N-Bus、TiGem、InnoVision、HUASAN、华三均为杭州华三通信技术有限公司的商标。对于本手册中出现的其它公司的商标、产品标识及商品名称，由各自权利人拥有。

除非另有约定，本手册仅作为使用指导，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。如需要获取最新手册，请登录http://www.h3c.com.cn。

技术支持

用户支持邮箱：customer_service@h3c.com

技术支持热线电话：800-810-0504（固话拨打）

400-810-0504（手机、固话均可拨打）

网址：http://www.h3c.com.cn

前 言

相关手册

手册名称	用途
H3C OAP Module User Manual	介绍了 OAP/OAPS 模块的使用方法。

本书简介

本手册各章节内容如下：

- **第 1 章 OAP 单板二次开发操作指导。**介绍 OAP 模块的二次开发指导说明。
- **第 2 章 OAPS 单板二次开发操作指导。**介绍 OAPS 模块的二次开发指导说明。

本书约定

1. 命令行格式约定

格　式	意　义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用**加粗**字体表示。
斜体	命令行参数（命令中必须由实际值进行替代的部分）采用*斜体*表示。
[ ]	表示用“[ ]”括起来的部分在命令配置时是可选的。
{ x | y | ... }	表示从两个或多个选项中选取一个。
[x | y | ...]	表示从两个或多个选项中选取一个或者不选。
{ x | y | ... }*	表示从两个或多个选项中选取多个，最少选取一个，最多选取所有选项。
[x | y | ...]*	表示从两个或多个选项中选取多个或者不选。
&<1-n>	表示符号&前面的参数可以重复输入 1～n 次。
#	由“#”号开始的行表示为注释行。

2. 各类标志

本书还采用各种醒目标志来表示在操作过程中应该特别注意的地方，这些标志的意义如下：

小心、注意：提醒操作中应注意的事项，不当的操作可能会导致数据丢失或者设备损坏。

警告：该标志后的注释需给予格外关注，不当的操作可能会对人身造成伤害。

说明、提示、窍门、思考：对操作内容的描述进行必要的补充和说明。

环境保护

本产品符合关于环境保护方面的设计要求，产品的存放、使用和弃置应遵照相关国家法律、法规要求进行。

目 录

第 1 章 OAP单板二次开发操作指导 1-1
1.1 OAP单板Linux 系统启动选项 1-1
1.2 Linux kernel和module的编译 1-2
第 2 章 OAPS单板二次开发操作指导 2-1
2.1 OAPS单板Linux 系统启动选项 2-1
2.2 Linux kernel和module的编译 2-2

第1章 OAP 单板二次开发操作指导

1.1 OAP 单板 Linux 系统启动选项

OAP 单板上的存储介质有 CF 卡（必选）和硬盘（可选）。CF 卡的 MBR 上安装有 Lilo。同时，CF 卡上保存了 Linux 系统启动所必需的 Kernel 文件和 initrd 文件。

注：OAP 单板只能支持 Lilo 作为系统的 bootloader，而无法支持 GRUB。

Lilo 启动时，提供用户两个引导选项，如下所示：

```
==============<BIOS-BASIC MENU(Ver 1.13)>====== =============
| <1> Boot Linux From CF  Card                                |
| <2> Install Linux OS  With Ethernet                         |
| <3> Backup BIOS to the backup portion                       |
| <4> Recover BIOS from the backup portion                    |
| <5> Reboot                                                  |
=============================================================
Enter your choice(1-5):1Reading boot sector...done.
Switching to boot loader.

LILO boot:
```

#（此处键入 tab 键，就可以看到两个启动选项）：

```
linux-hdc       linux-ramdisk
boot:
```

#（在此处输入两个启动选项中的任意一个，即可启动相应的系统）

1. Linux-ramdisk 选项

将 ramdisk 作为根文件系统，将 CF 卡的分区 mount 为/boot 分区。此系统启动后，提供了私有协议模块、看门狗、CPLD 驱动等基本功能。

2. Linux-hdc 选项：

将硬盘作为根文件系统启动。硬盘上安装了一个比较完整的根文件系统，提供了用户二次开发所必须的常用工具，诸如 gcc、rpm 与 vi 编辑器。也启用了一些网络服务功能。用户完全可以象配置一台 PC 机一样配置 OAP 单板上 Linux 操作系统的服务。系统默认启动此配置。

出厂时，硬盘和 CF 卡的分区情况如下表所示：

表1-1 OAP 单板存储设备默认分区方式

设备名	分区名	挂载点	文件系统格式	功能
CF 卡	/dev/hda1	/boot	ext3	Linux 系统的 boot 分区，MBR 中存放有 lilo，Linux 系统启动所必需的 kernel、initrd 文件。此分区卷标为 /boot
硬盘	/dev/hdc1	/	ext3	Linux 系统的根文件系统，当从 Lilo 中选择使用 linux-hdc 启动时，将此分区作为系统的根文件系统。此分区的卷标为/。
	/dev/hdc2	/alt	ext3	作为根文件系统的备份分区，出厂时为空。
	/dev/hdc3	/opt	ext3	作为最大的一个分区，供用户使用。没有保存任何系统文件。
	/dev/hdc4	swap	swap	Linux 系统的交换，当从 Lilo 中选择使用 linux-hdc 启动时，将此分区作为系统的交换分区。

用户可以根据自己的需求来重新定制分区方式，但是需要注意的是，由于 BIOS 的特殊性，只能将 CF 卡的第一个分区作为系统的 boot 分区。

1.2 Linux kernel 和 module 的编译

当用户选择的引导选项为 linux-hdc，系统启动后将硬盘的/dev/hdc1 分区作为根文件系统。此时，用户可以基于 Linux 系统进行二次开发。除了输入/输出方式的不同之外，这样的开发形式与在一台安装了 Linux OS 的 PC 机上开发没有区别。

1. Linux 内核的编译

但是，如果用户有特殊需求，比如，需要重新配置内核的特性，或者是需要编译一些可加载的设备驱动模块，就必须重新编译内核。

OAP 单板在出厂时，并没有附带 2.6.9-42.0.3.EL 的源码。可以从网站上下载这个版本的内核源码，下载地址如下：

ftp://updates.redhat.com/enterprise/4AS/en/os/SRPMS/kernel-2.6.9-42.0.3.EL.src.rpm。

但是由于 OAP 单板硬件的特殊性，此源码并不适用于 OAP 单板，必须打上特定的补丁程序之后，才可以编译适用于 OAP 单板的内核源代码，此补丁程序如下：

```
kernel-2.6.9-42.0.3.EL-appboard.patch
```

建议用户按照如下方式操作，进行内核的编译

(1) 将内核的源码下载到/opt 目录下

```
[root@localhost root]#cd /opt
```

```
[root@localhost                                                    opt]#scp
root@192.168.0.2:/home/zxh/centos/kernel-2.6.9-42.0.3.EL.src.rpm ./
root@192.168.0.2's password:
kernel-2.6.9-42.0.3.EL.src.rpm           100%   45MB  11.3MB/s   00:04
[root@localhost opt]#
```

(2) 下载补丁程序

```
[root@localhost                          opt]#                          scp
root@192.168.0.2:/home/zxh/centos/kernel-2.6.9-42.0.3.EL-appboard.patch ./
root@192.168.0.2's password:
kernel-2.6.9-42.0.3.EL-appboard.patch    100%   14KB  13.8KB/s   00:00
[root@localhost opt]#
```

(3) 解压内核源代码

```
[root@localhost opt]# rpm -Uvh kernel-2.6.9-42.0.3.EL.src.rpm
[root@localhost opt]# cd /usr/src/redhat/SPECS/
[root@localhost SPECS]# rpmbuild -bp --target $(uname -m) kernel-2.6.spec
```

(4) 给初始源码打补丁，生成适用于 OAP 单板的内核源码

```
[root@localhost SPECS]#cd ../BUILD/kernel-2.6.9/linux-2.6.9/
[root@localhost                                              linux-2.6.9]#cp
/opt/kernel-2.6.9-42.0.3.EL-appboard.patch ./
[root@localhost          linux-2.6.9]#          patch          -p1          <
kernel-2.6.9-42.0.3.EL-appboard.patch
patching file arch/i386/boot/video.S
patching file arch/i386/kernel/reboot.c
patching file drivers/net/e1000/e1000_hw.c
patching file drivers/serial/8250.c
patching file init/main.c
patching file Makefile
[root@localhost linux-2.6.9]#
```

(5) 开始编译内核以及驱动模块

```
[root@localhost linux-2.6.9]# make clean
[root@localhost linux-2.6.9]# make mrproper
[root@localhost linux-2.6.9]# cp configs/kernel-2.6.9-i686.config ./.config
[root@localhost linux-2.6.9]# make oldconfig
[root@localhost linux-2.6.9]# make menuconfig
```

此时，用户可以根据自己的需要定制内核的特性

```
[root@localhost linux-2.6.9]# make
[root@localhost linux-2.6.9]# make modules
[root@localhost linux-2.6.9]# make modules_install
[root@localhost linux-2.6.9]# make installl
```

2. lilo 配置

在编译好新的内核和 initrd 之后，用户需要手动的刷新 lilo，以使得系统可以使用新的内核引导。

Lilo 的配置文件 lilo.conf 放在/etc 目录下，使用 vi 编辑器可以修改其中的启动设置。lilo.conf 文件的内容如下：

```
prompt
timeout=50
default=linux-hdc
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
linear
image=/boot/vmlinuz-2.6.9-42.0.3.EL
         label=linux-ramdisk
         initrd=/boot/initrd-2.6.9-42.0.3.EL-appboard.img
         append="root=/dev/ram0 console=ttyS0,9600 rw"
image=/boot/vmlinuz-2.6.9-42.0.3.EL
         label=linux-hdc
         initrd=/boot/initrd-2.6.9-42.0.3.EL.img
         append="console=ttyS0,9600 root=LABEL=/ "
```

需要注意的是 ，由于 OAP 单板系统启动的信息输入都是通过串口进行的，所以需要通过 lilo 追加以下的参数：

```
append="console=ttyS0,9600"
```

如果使用超级终端软件连接到主机的 console，再使用 oap connect slot x 命令连接到 OAP 单板的串口上，请将超级终端的终端仿真设置为 VT100，这样 vi 编辑器就可以正常使用了。如图所示：

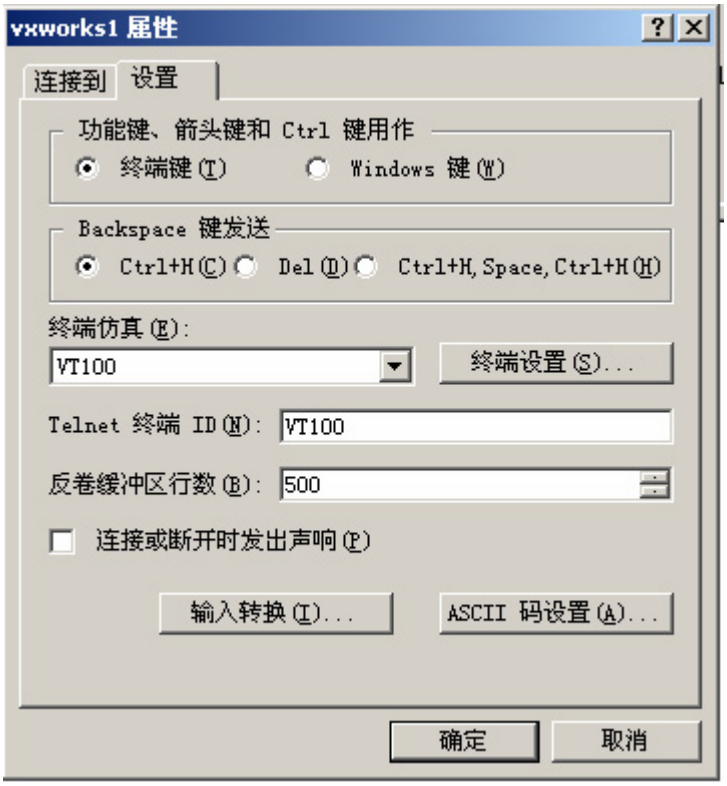

图1-1 终端类型设置

3. e1000 驱动编译

OAP 单板上使用了 Intel 的千兆以太网芯片，默认提供的 e1000 的驱动程序的版本是 7.0.33。在给内核打补丁的同时，同时也对 e1000 的代码进行了修改。如果用户有需求，需要更新以太网驱动，那么首先要从 Intel 的官方网站上下载特定版本的 e1000 驱动，然后打上 OAP 单板的以太网驱动的补丁之后，才能够编译出适用于 OAP 单板以太网的驱动程序。

以 e1000 驱动 7.0.41 为例

(1) 从 Intel 官方网站获取了 e1000 驱动 7.0.41 的源码 e1000-7.0.41.tar.gz 后，进行解压缩。

```
[root@localhost tmp]# ls
e1000-7.0.41.tar.gz
[root@localhost tmp]# tar -zxf e1000-7.0.41.tar.gz -C ./
[root@localhost tmp]#
```

(2) 获得 e1000 驱动对应的 patch 文件并复制到特定的目录下

```
[root@localhost tmp]# cd e1000-7.0.41
[root@localhost e1000-7.0.41]# ls
e1000.7                      e1000.spec    LICENSE      README  SUMS
e1000-7.0.41-appboard.patch  ldistrib.txt  pci.updates  src
[root@localhost e1000-7.0.41]# patch -p1 < e1000-7.0.41-appboard.patch
```

```
patching file src/e1000.h
patching file src/e1000_hw.c
patching file src/e1000_main.c
[root@localhost e1000-7.0.41]#
```

(3) 编译 e1000 的驱动程序

```
[root@localhost e1000-7.0.41]#cd src
[root@localhost src]#make clean
[root@localhost src]#make
[root@localhost src]#make install
```

(4) 然后重新引导系统，确认 e1000 的驱动以及升级为指定的版本

```
[root@localhost root]# ethtool -i eth0
driver: e1000
version: 7.0.41
firmware-version: N/A
bus-info: 01:00.0
[root@localhost root]#
```

对于其他模块的编译没有特殊性，手册中也不再进行详细描述。

第2章 OAPS 单板二次开发操作指导

2.1 OAPS 单板 Linux 系统启动选项

OAPS 单板上的存储介质有 CF 卡和硬盘。CF 卡的 MBR 上安装有 Lilo，同时，CF 卡上保存了 Linux 系统启动所必需的 Kernel 文件和 initrd 文件。

注：OAPS 单板可同时支持 Lilo 和 Grub 作为系统的 bootloader。

Lilo 启动时，提供用户两个引导选项，如下所示：

```
============<BIOS-BASIC MENU(Ver1.00)>==================
| <1> Boot Linux from CF Card                            |
| <2> Boot Linux from Hard Disk                          |
| <3> Boot Linux from USB                                |
| <4> Boot Linux from Ethernet                           |
| <5> Default boot device                                |
| <6> Backup BIOS to the backup portion                  |
| <7> Recover BIOS from the backup portion               |
| <8> Reboot                                             |
=========================================================
Enter your choice(1-8):1
Reading boot sector...done.
Switching to boot loader.

LILO boot:
```

#（此处键入 tab 键，就可以看到两个启动选项）：

```
linux-hdc       linux-ramdisk
boot:
```

#（在此处输入两个启动选项中的任意一个，即可启动相应的系统）

1. Linux-ramdisk 选项

将 ramdisk 作为根文件系统，将 CF 卡的分区 mount 为/boot 分区。此系统启动后，提供了私有协议模块、看门狗、CPLD 驱动等基本功能。

2. Linux-hdc 选项：

将硬盘作为根文件系统启动。硬盘上安装了一个比较完整的根文件系统，提供了用户二次开发所必须的常用工具，诸如 gcc、rpm 与 vi 编辑器。也启用了一些网络服务功能。用户完全可以象配置一台 PC 机一样配置 OAPS 单板上 Linux 操作系统的服务。系统默认启动此配置。

出厂时，硬盘和 CF 卡的分区情况如下表所示：

表2-1 OAPS 单板存储设备默认分区方式

设备名	分区名	挂载点	文件系统格式	功能
CF 卡	/dev/hda1	/boot	ext3	Linux 系统的 boot 分区，MBR 中存放有 lilo，Linux 系统启动所必需的 kernel、initrd 文件。此分区卷标为/boot
硬盘	/dev/sda1	/	ext3	Linux 系统的根文件系统，当从 Lilo 中选择使用 linux-hdc 启动时，将此分区作为系统的根文件系统。此分区的卷标为/。
	/dev/sda2	/alt	ext3	作为根文件系统的备份分区，出厂时为空。
	/dev/sda3	/opt	ext3	作为最大的一个分区，供用户使用。没有保存任何系统文件。
	/dev/sda4	swap	swap	Linux 系统的交换，当从 Lilo 中选择使用 linux-hdc 启动时，将此分区作为系统的交换分区。

注：用户可以根据自己的需求来重新定制分区方式。

2.2 Linux kernel 和 module 的编译

当用户选择从 CF 卡启动时旋子的启动选项为 linux-hdc，或者用户直接选择从硬盘启动时，系统启动后会将硬盘的/dev/sda1 分区作为根文件系统。此时，用户可以基于 Linux 系统进行二次开发。除了输入/输出方式的不同之外，这样的开发形式与在一台安装了 Linux OS 的 PC 机上开发没有区别。

1. Linux 内核的编译

如果因为特殊功能的需要要重新编译内核，用户可以直接从下面链接中下载内核，并且重新配置内核，下载地址如下：

ftp://updates.redhat.com/enterprise/4AS/en/os/SRPMS/kernel-2.6.9-42.0.3.EL.src.rpm。

下载完内核后，用户可以按照下面描述的操作，解压并编译内核：(假设内核的源码下载到/opt 目录下)

1> 解压内核源代码

```
[root@localhost root]#cd /opt
[root@localhost opt]# rpm -Uvh kernel-2.6.9-42.0.3.EL.src.rpm
[root@localhost opt]# cd /usr/src/redhat/SPECS/
[root@localhost SPECS]# rpmbuild -bp --target $(uname -m) kernel-2.6.spec
```

2> 配置内核

```
[root@localhost linux-2.6.9]# make clean
```

```
[root@localhost linux-2.6.9]# make mrproper
[root@localhost linux-2.6.9]# make menuconfig
```

2> 配置内核和模块

```
[root@localhost linux-2.6.9]# make
[root@localhost linux-2.6.9]# make modules
[root@localhost linux-2.6.9]# make modules_install
[root@localhost linux-2.6.9]# make installl
```

2. Lilo 配置

编译好新的内核和 initrd 之后，用户需要手动的刷新 Lilo，使得系统可以使用新的内核引导。

Lilo 的配置文件 lilo.conf 放在/etc 目录下，使用 vi 编辑器可以修改其中的启动设置。

Lilo.conf 文件的内容如下：

```
prompt
timeout=50
default=linux-hdc
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
linear
image=/boot/vmlinuz-2.6.9-42.0.3.EL-oaps
         label=linux-ramdisk
         initrd=/boot/initrd-2.6.9-42.0.3.EL-appboard-oaps.img
         append="root=/dev/ram0 console=ttyS0,9600 rw"
image=/boot/vmlinuz-2.6.9-42.0.3.EL-oaps
         label=linux-hdc
         initrd=/boot/initrd-2.6.9-42.0.3.EL-oaps.img
         append="console=ttyS0,9600 rw root=LABEL=/"
```

需要注意的是，由于 OAPS 单板系统启动的信息输入都是通过串口进行的，所以需要通过 Lilo 追加以下的参数：

```
append="console=ttyS0,9600"
```

如果使用超级终端软件连接到主机的 console，再使用 oap connect slot x 命令连接到 OAPS 单板的串口上，请将超级终端的终端仿真设置为 VT100，这样 vi 编辑器就可以正常使用了。如图所示：

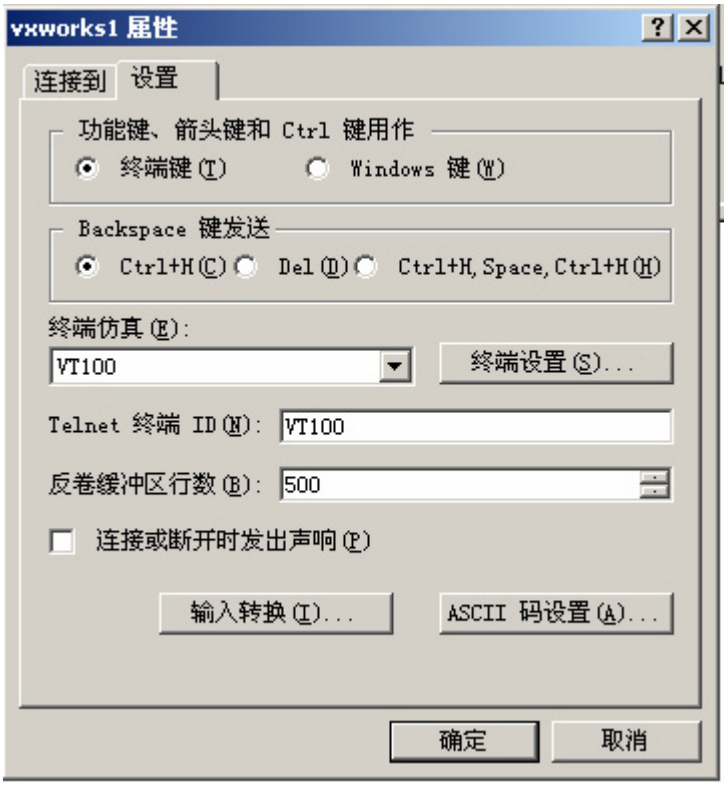
vxworks1 属性
连接到
设置
功能键、箭头键和 Ctrl 键用作
终端键(T)
Windows 键(W)
Backspace 键发送
Ctrl+H(C)
Del(D)
Ctrl+H, Space, Ctrl+H(H)
终端仿真(E):
VT100
终端设置(S)...
Telnet 终端 ID(N):
VT100
反卷缓冲区行数(B):
500
连接或断开时发出声响(P)
输入转换(I)...
ASCII 码设置(A)...
确定
取消

图2-1 终端类型设置